

Nama : Mufli Agnal Noorshodiq

Kelas : 2IA11

NPM : 50422926

## **TUGAS PBO**

### **MERANGKUM PERTEMUAN 1 – 6**

PERT 1-2

#### **PENGENALAN PEMROGRAMAN BERBASIS OBJEK**

Pemrograman berorientasi objek atau yang biasa disebut OOP adalah paradigma pemrograman yang menggunakan objek sebagai unit utama. Tujuannya adalah untuk mengatasi kompleksitas masalah program dengan menyusun kode ke dalam entitas yang memiliki atribut dan perilaku. Meskipun konsep OOP mulai mendapatkan perhatian pada tahun 1980, pengembangan bahasa seperti simula dan smalltalk pada tahun 1960 dan 1970 berperan penting dalam pengembangan konsep ini.

Pemrograman berorientasi objek (OOP) tidak fokus pada struktur pemecahan masalah, melainkan pada objek-objek yang dapat menyelesaikannya. OOP efektif apabila digunakan bersama dengan Objek-Oriented Analysis dan Design Process (OOAD).

Perbedaan pemrograman berbasis objek dan pemrograman terstruktur adalah pemrograman berbasis objek (OOP) menggunakan objek sebagai unit utama, sementara pemrograman terstruktur dengan menggunakan struktur seperti pernyataan if, loop, dan fungsi. OOP lebih fokus pada objek yang memiliki atribut dan perilaku untuk memecahkan masalah, sementara pemrograman terstruktur lebih terfokus pada langkah-langkah algoritma.

Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar. Sebagai analogi, class bisa diibaratkan dengan kendaraan. Kita tahu bahwa kendaraan memiliki ciri-ciri seperti merk, memiliki mesin, memiliki ban, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah kendaraan. Selain memiliki ciri-ciri, sebuah kendaraan juga bisa dikenakan tindakan, seperti: menghidupkan mesin atau mematikan mesin.

Objek dalam pemrograman adalah entitas yang unik yang merepresentasikan sesuatu dari dunia nyata, seperti orang, mobil, atau rekening bank. Objek memiliki atribut dan tingkah laku. Misalnya, objek sepeda memiliki atribut seperti pedal, roda, dan warna. Dalam pengembangan perangkat lunak berorientasi objek, atribut objek disimpan dalam variabel dan tingkah laku disimpan dalam metode. Pembuatan objek melibatkan langkah-langkah deklarasi variabel, instansiasi objek baru, dan pemanggilan konstruktor dengan kata kunci new.

## PERT 3-6

### INHERITANCE, ENCAPSULATION DAN POLYMORPHISM

Inheritance adalah konsep dalam pemrograman berbasis objek di mana sebuah class baru dapat dibuat dari class yang sudah ada, dengan menambahkan atau memodifikasi fitur-fiturnya. Class yang baru ini disebut subclass atau child class, sementara class yang digunakan sebagai dasar untuk subclass tersebut disebut superclass atau parent class. Inheritance memungkinkan untuk menggunakan kembali kode dan memperluas fungsionalitas class yang sudah ada. Dalam inheritance tunggal, sebuah subclass hanya dapat mewarisi dari satu superclass, sedangkan dalam inheritance ganda, sebuah subclass dapat mewarisi dari dua atau lebih superclass, namun Java tidak mendukung inheritance ganda. Konstruktor tidak diwariskan secara otomatis ke subclass, kecuali jika menggunakan perintah 'super'. Untuk menerapkan inheritance dalam Java, digunakan kata kunci "extends".

Encapsulation adalah konsep dalam pemrograman berorientasi objek yang bertujuan melindungi atribut atau metode tertentu dalam sebuah kelas agar tidak dapat diakses atau dimodifikasi secara sembarangan oleh bagian program lain. Ini dilakukan dengan menggunakan access modifiers (hak akses) seperti default, public, protected, dan private. Accessors adalah metode untuk mengakses atribut atau properti yang telah dilindungi, sehingga memungkinkan akses terkontrol terhadap data dalam sebuah objek.

Overloading adalah kemampuan dalam pemrograman berorientasi objek di mana sebuah class dapat memiliki lebih dari satu method dengan nama yang sama, tetapi memiliki parameter yang berbeda.

Overriding adalah kemampuan subclass untuk memodifikasi method dari superclass-nya dengan mendefinisikan kembali method yang sama dengan nama dan parameter yang identik. Contohnya, subclass "Mobil" dapat menumpuk (override) method "keterangan()" dan "hapus()" yang ada pada superclass "Kendaraan".

